ФПМИ БГУ

Вычислительные методы алгебры
Лаборатоная работа 3

Решение СЛАУ методом левой прогонки

Подготовил:
Ткачук Павел
2 курс 1 группа

Преподаватель:
Будник Анатолий Михайлович

20 декабря 2016 г.

1 Постановка задачи

Система:

$$\begin{cases} c_0 y_0 - b_0 y_1 = f_0 \\ -a_i y_{i-1} + c_i y_i - b_i y_{i+1} = f_i,\ i = \overline{1, n-1}. \\ -a_n y_{n-1} + c_n y_n = f_n. \end{cases} \tag{1}$$

Входные данные:

$$\left[\begin{array}{ccccc|c} 0.6444 & 0.0000 & 0 & 0 & 0 & 1.2677 \\ -0.0395 & 0.4208 & 0.0000 & 0 & 0 & 1.6819 \\ 0 & -0.1184 & 0.7627 & 0.0145 & 0 & -2.3657 \\ 0 & 0 & 0.0000 & -0.0960 & 0.7627 & -6.5369 \\ 0 & 0 & 0 & 0.1907 & 0.5523 & 2.8351 \end{array}\right]$$

Задача:

1. Методом левой прогонки найти решение СЛАУ x
2. Найти вектор невязки $r = Ax - b$

2 Алгоритм

1. Вычисляем прогоночные коэфициенты

$$\xi_n = \frac{a_n}{c_n},\ \xi_i = \frac{a_i}{c_i - \xi_{i+1} b_i},\ i = n-1, n-2, \ldots, 1$$
$$\eta_n = \frac{f_n}{c_n},\ \eta_i = \frac{f_i + b_i \eta_{i+1}}{c_i - b_i \xi_{i+1}},\ i = n-1, n-2, \ldots, 0$$

2. Выполняем обратный ход

$$y_0 = \eta_0,\ y_{i+1} = \xi_{i+1} y_i + \eta_{i+1},\ i = 1, 2, \ldots, n-1$$

3 Результаты и вывод

3.1 Входные данные

0.6444 0.0000 0 0 0 1.2677
-0.0395 0.4208 0.0000 0 0 1.6819
0 -0.1184 0.7627 0.0145 0 -2.3657
0 0 0.0000 -0.0960 0.7627 -6.5369
0 0 0 0.1907 0.5523 2.8351

3.2 Выходные данные

```
Решаем матричное уравнение методом прогонки
Основная матрица системы:
[[ 0.6444  0.      0.      0.      0.    ]
 [-0.0395  0.4208  0.      0.      0.    ]
 [ 0.     -0.1184  0.7627  0.0145  0.    ]
 [ 0.      0.     -0.096   0.7627  0.    ]
 [ 0.      0.      0.     -0.0158  0.5523]]
Свободные члены:  [ 1.2677  1.6819 -2.3657 -6.5369  2.8351]
Решение системы:  [ 1.96725636  4.18157468 -2.28419695 -8.85824427  4.87984744]
Невязка:  [ 0.  0.  0.  0.  0.]
Норма невязки:  0.0
```

3.3 Вывод

Вектор невязки нулевой, значит результат, выданный программой, являеться достаточно точным. Отметим, что данный метод обладает сложностью $O(n)$ - это дает значительный выигрыш по времени выполнения, в сравнении с методом Гаусса (его сложность $O(n^3)$). Однако метод прогонки применим только для матриц трехдиагонального вида, в отличии от метода Гаусса, который выдает решение любой совместной системы линейных уравнений.

4 Листинг кода

```
import numpy as np
import numpy.linalg as linalg
# Метод левой прогонки
def solve(matr_A, matr_f):
    A = np.array(matr_A)
    f = np.array(matr_f)
    size = len(f)
    b = np.zeros(size)
    a = np.zeros(size)
    c = np.zeros(size)
    for i in range(size):
        c[i] = A[i, i]
        if i != size - 1:
            b[i] = -A[i, i + 1]
        if i != 0:
            a[i] = -A[i, i - 1]
    ksi = np.zeros(size + 1)
    for i in reversed(range(size)):
        ksi[i] = (a[i]) / (c[i] - ksi[i + 1] * b[i])
    eta = np.zeros(size + 1)
    for i in reversed(range(size)):
        eta[i] = (f[i] + b[i] * eta[i + 1]) / (c[i] - ksi[i + 1] * b[i])
    y = np.zeros(size)
    for i in range(size):
        y[i] = ksi[i] * y[i - 1] + eta[i]
    return y
file = open("matrix", "r")  # Чтение файла
A, b = [], []
for line in file:
    A.append([float(el) for el in line.split()[:-1]])
    b.append(float(line.split().pop()))
A = [[A[i][j] if np.abs(i - j) < 2 else 0 for j in range(len(b))] for i in range(len(b))]
A = np.array(A)
b = np.array(b)
print("Решаем матричное уравнение методом прогонки")
print("Основная матрица системы:")
print(A)
print("Свободные члены: ", b)
ans = solve(A, b)
print("Решение системы: ", ans)
print("Невязка: ", np.dot(A, ans) - b)
print("Норма невязки: ", linalg.norm(np.dot(A, ans) - b))
```